VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Univerzální administrační systém farnosti, část 2
Parish universal administration system, part 2

2010

Michael Hrabálek

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě dne 7. 5. 2010 ….....….....

Michael Hrabálek

Abstrakt

Práce je druhou částí projektu zabývajícím se vývojem aplikace pro potřeby webové prezentace farností a děkanátů. Popisuje zejména systém pro vytváření šablon grafického uživatelského rozhraní a fotogalerii spolu s appletem pro nahrávání fotografií. Rozebírá technologie, které se k danému problému naskytují, jejich výhody a nevýhody. Také jsou popsány problémy a jejich řešení, které se během vývoje vyskytly.

Klíčová slova

JavaScript, Ajax, CSS, HTML, Java, Java Applet, PHP, internet, web, aplikace, jQuery, XML, šablona, systém, fotogalerie, gui, imagebox

Abstract

The work describes the development of a web application for the needs of a parish, which is intended to create a simple website. It primarily describes a system for creating templates of graphical user interface and photogallery with an applet to uploading photos. Discusses technologies solving this issues, their advantages and disadvantages. It also describes the problems and their solutions that occurred during development.

Key words

JavaScript, Ajax, CSS, HTML, Java, Java Applet, PHP, internet, web, application, jQuery, XML, template, system, photogallery, gui, imagebox

Seznam použitých symbolů a zkratek

AJAX	Asynchronous JavaScript And XML
API	Application Programming Interface
CSS	Cascading Style Sheet
DOM	Document Object Model
HTML	Hyper Text Markup Language
HTTP	Hyper Text Transfer Protokol
JDK	Java Development Kit
JRE	Java Runtime Enviroment
JSON	JavaScript Object Notation
JVM	Java Virtual Machine
OOP	Object Oriented Programming
PDF	Portable Document Format
PHP	PHP Hypertext Preprocesor
SQL	Structured Query Language
URL	Uniform Resource Locator
W3C	World Wide Web Consortium
XML	Extensible Markup Language

Obsah

1 Úvod

1.1 Cíl práce

Cílem práce je vývoj webové aplikace pro potřeby farností v České republice, především pak pro server www.farnost.cz, který poskytuje hosting a doménu pro stránky farností. Práce úzce souvisí s bakalářskou prací studenta Jindřicha Kuchaře. Tato část je soustředěna na tvorbu vzhledu, fotogalerii, instalaci a uživatelskou dokumentaci.

1.2 Proč nový systém

Žijeme v době, kdy téměř každá instituce, organizace a jiná seskupení, prezentuje své vlastní dění na internetu formou webové prezentace. V církvi funguje hierarchické uspořádání mnoha úřadů. Od farnosti, jakožto nejmenšího článku počínaje, přes děkanát, biskupství, arcibiskupství a papežským úřadem konče. Avšak jen od biskupského úřadu výše, si mohou dovolit zaměstnance pro tvorbu webu a dalších IT funkcí. Sice se najde mnoho dobrovolníků, kteří se do tvorby webu zapojují, bohužel ne každá farnost/děkanát disponuje takovými lidmi. Proto je zapotřebí vyvinout systém s jednoduchým nasazením a použitím, pomocí kterého si každá farnost/děkanát vytvoří webovou prezentaci a to bez jakékoli znalosti v oboru.

1.3 Kdo se systémem bude pracovat

Se systémem budou pracovat především lidé bez znalosti tvorby webu. Proto je nutné, aby byl uživatelsky přívětivý. Může s ním však pracovat i člověk se znalostmi tvorby webu a proto je dobré, aby byl systém zdokumentován a schopen případných úprav.

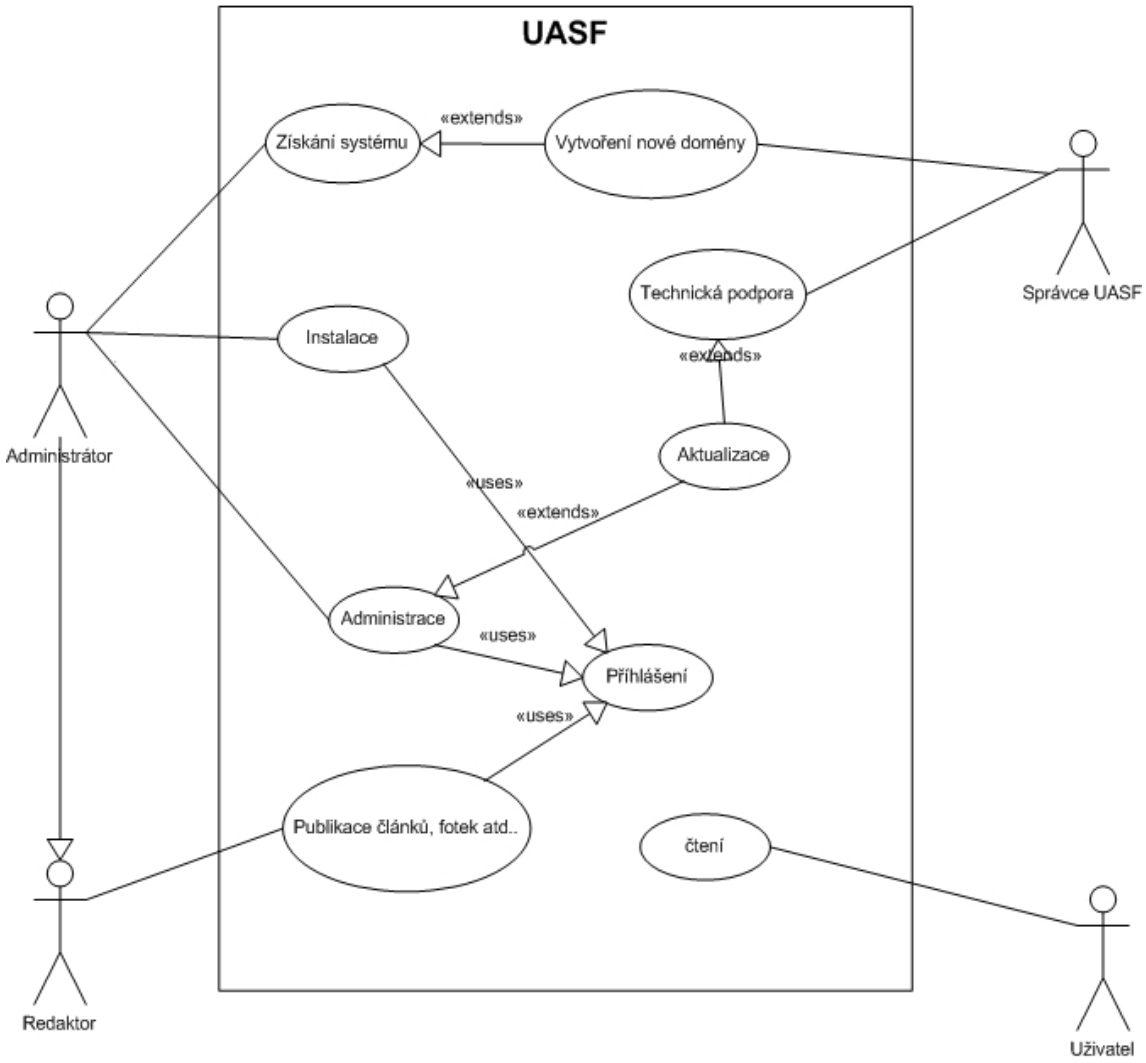
1.4 O systému

Základní myšlenkou je vyvinout systém, který se jednoduše nainstaluje, jednoduše udržuje a přitom by měl mít možnost v každé jeho instanci jiný vzhled. Taky je potřeba aby obsahoval funkce potřebné pro oblast pro níž je vyvíjen. Musí být zároveň schopen být doplněn o další funkcionalitu. Aby bylo možné takový systém vyvinout, byl kladen velký důraz na výběr technologií. V první části práce je zmíněn výběr technologií pro programování na straně serveru a výběr SŘBD. Zde se budeme především zabývat výběrem technologií pro uživatelské rozhraní a stranu klienta.

Když se rozhlídneme nad dnešním trhem, jistě najdeme několik systémů s podobnou myšlenkou. Základním problémem může být ale cena, jelikož jedním z požadavků jsou velmi nízké náklady. Existují však i systémy poskytované zcela zdarma. U nich ale také

můžeme narazit na problémy. Většinou totiž vyžadují alespoň minimální znalosti programování a pro použití je mnohdy třeba nastudovat dlouhou dokumentaci.

Pro lepší představu si ukážeme systém na diagramu případu užití.



Obrázek 1.1: Diagram případů užití systému UASF

Jak vidíme na obrázku, tak základním rozdílem oproti nabízejícím se systémům je možnost vlastnit systém, aniž bychom museli zařizovat vlastní doménu či hosting. I když je toto bezesporu velká výhoda, její implementace je již nad rámec této práce a proto ji více prostoru věnováno nebude.

V práci se máme ale zabývat vzhledem, fotogalerií, instalací a uživatelské dokumentaci. Z uvedených případů užití se to tedy týká instalace, administrace, publikace a čtení.

1.5 Struktura práce

Nejprve se zaměříme na popis řešení tvorby univerzálního vzhledu a technologie s ním spojené. V další části si popíšeme jakým způsobem byla vyřešena fotogalerie. A nakonec si řekneme něco o instalaci.

2 Tvorba vzhledu

2.1 Požadavky

- Možnost odlišného vzhledu v každé instanci aplikace
- Jednoduché navržení struktury a vzhledu stránky
- Musí vypadat stejně na nejpoužívanějších prohlížečích
- Stránky musí být validní

Než si popíšeme samotný systém tvorby vzhledu, který byl pro tuto aplikaci navržen, probereme si technologie, které systém využívá.

2.2 XHTML

XHTML neboli Extensible HyperText Markup Language je typ dokumentu, který využívá, vychází a rozšiřuje dokumenty HTML 4 a zároveň je založen na typu dokumentu XML.

2.2.1 HTML 4

HTML je značkovací jazyk pro vytváření dokumentů především v systému World Wide Web. Jeho počátky sahají už do roku 1989, kdy byl vyvinut jazyk SGML. Samotné HTML bylo však navrženo až o rok později. Od té doby prošel značným vývojem a standardizací, která pokračuje dodnes. V současné době se pracuje na vývoji HTML 5. Základem jeho specifikace má být Web Application 1.0 a Web Forms 2.0. Specifikace by měla být hotová do roku 2012. Avšak úplné dokončení s opravami problémů, které se ukážou během provozu je plánováno až na rok 2022.

V XHTML se však ale využívá specifikace verze 4, která byla vydána již v roce 1997 pod záštitou organizace W3C.

2.2.2 XML

XML je obecný značkovací jazyk, který byl rovněž standardizován konsorciem W3C. Na rozdíl od HTML umožňuje vytvářet vlastní značkovací jazyk, sloužící různým účelům. Je také hojně používán pro ukládání a výměnu dat. Další z jeho výhod je možnost transformovat jej jazykem XSL do jiných XML aplikací.

2.2.3 Výhody

Jak již bylo zmíněno, XHTML je kombinací HTML 4 a XML, tudíž nese všechny výhody obou technologií. Umožňuje tedy používat jak značek HTML, tak definovat značky nové a používat další nástroje pro práci s XML.

2.3 CSS

CSS neboli Cascading Style Sheets, v češtině kaskádové styly jsou jednoduchým jazykem nebo také mechanismem pro grafickou úpravu webových dokumentů. Především jsou tedy využívány pro dokumenty HTML, XML a tedy i XTHML.

2.3.1 Vznik

Na počátku 90. let se začaly objevovat dokumenty HTML. V té době se ale ještě nevyužívaly v takovém rozsahu, jako je známe dnes. Také systém World Wide Web byl ještě v plenkách. S postupným rozmachem WWW a internetu, však množství webů ohromně stouplo a došlo také na potřebu graficky upravit vzhled dokumentů. Tvůrci HTML tedy přinesli nové značky a prvky, kterýma se zabývalo především HTML ve verzi 3.2. Tím byla ale spojena struktura dokumentu s jeho vzhledem.

Proto se začal vyvíjet mechanismus, který by oddělil vzhled dokumentu od jeho obsahu. První řešení se objevilo v roce 1994 s názvem Cascading Style Sheet. Jeho první standardizovaná verze byla uveřejněna však až o dva roky později.

Standardizaci HTML a CSS nejprve řešila jedna skupina lidí. Později však byla rozdělena, což vedlo k vyústěním mnoha změn a specifikaci CSS 2. To bylo v roce 1997. Hned po vydání CSS 2 začala práce na CSS 3, které je ale dodnes ve vývoji.

2.3.2 Problémy

Největším problémem kaskádových stylů byla jejich integrace do webových prohlížečů. O to se největší měrou zasloužila firma Microsoft, jelikož dodnes zabírá se svým prohlížečem Internet Explorer více jak polovinu trhu a implementace standardů jim trvá o několik let déle než ostatním prohlížečům na trhu. Například, i když byla v roce 1996 vydána první verze CSS a Internet Explorer ve verzi 3 následoval chvíli po ní, implementace CSS byla jen částečná. Plná podpora CSS 1 se objevila v Internet Exploreru až ve verzi 5 v roce 2000, kdy už byla dva roky vydána nová specifikace CSS 2.

Dalším problémem bylo a dodnes je odlišné propočítávání například šířek okrajů, vnějšího a vnitřního odsazení prvků apod.. Aby stránky vypadaly ve všech prohlížečích stejně musí se dodnes využívat různých kliček. Jednou z prvních bylo napsání zvláštních stylů pro

každý prohlížeč, které se pak pomocí schopností JavaScriptu zobrazily. Toto řešení, zvláště dříve, naráželo na problém ne zcela velké rozšířenosti JavaScriptu. Postupem času se však JavaScript hojně rozšířil, ale pořád bylo nutné psát jednu věc vícekrát.

Později byla objevena další klička. A to taková, že prohlížeče, které zobrazovaly prvky nestandardně naštěstí zároveň neměly implementovánu dědičnost kaskádových stylů. Stačilo tedy v zděděné části styl nastavit na jiný. Ukažme si to na konkrétním případu.

```
.nadpis {
  width: 860px;
  margin-bottom: -4px;
  text-align: center;
  letter-spacing: 1px;
}

*>.nadpis {
  margin-bottom: 0px;
}
```

Takto definované odsazení nějakého prvku na stránce by tedy vypadalo shodně ve všech prohlížečích. Bohužel však v této možnosti pořád musíme psát určitý kód vícekrát a navíc nic netrvá věčně a dědičnost byla implementována i do nestandardně zobrazujících prohlížečů.

2.3.3 Reset CSS

Naštěstí se objevilo mnohem čistší řešení, které vyřeší mnoho problémů kompatibility zobrazení stránek prohlížeči. Obecně je známo jako Reset CSS. Vše okolo CSS Resetu bychom našli na stránkách www.css-reset.com nebo na stránkách Erica Meyrse, který je i autorem několika knih o css. V této práci byl CSS Reset použit a proto si jeho řešení ukažme.

```
html, body, div, span, applet, object, iframe,
h1, h2, h3, h4, h5, h6, p, blockquote, pre,
a, abbr, acronym, address, big, cite, code,
del, dfn, em, font, img, ins, kbd, q, s, samp,
small, strike, strong, sub, sup, tt, var,
b, u, i, center,
dl, dt, dd, ol, ul, li,
fieldset, form, label, legend,
table, caption, tbody, tfoot, thead, tr, th, td {
        margin: 0;
        padding: 0;
        border: 0;
        outline: 0;
        font-size: 100%;
        vertical-align: baseline;
        background: transparent;
}
```

```
body {
        line-height: 1;
}
ol, ul {
        list-style: none;
}
blockquote, q {
        quotes: none;
}
blockquote:before, blockquote:after,
q:before, q:after {
        content: '';
        content: none;
}
:focus {
        outline: 0;
}
ins {
        text-decoration: none;
}
del {
        text-decoration: line-through;
}
table {
        border-collapse: collapse;
        border-spacing: 0;
}
```

2.4 JavaScript

2.4.1 Úvod

Javascript je interpretovaný programovací jazyk se základními objektově orientovanými schopnostmi. Byl vyvinut firmou Netscape v roce 1995. Od té doby prošel mnohými změnami a v dnešní době je standardizován asociací ECMA. Někdy je proto také nazýván ECMAScript. Nejčastěji je používán pro práci s WWW stránkou. Nachází ale také využití i v jiných aplikacích než je webový prohlížeč. Nejznámějším příkladem je Adobe Acrobat.

2.4.2 Základní možnosti

- Řízení vzhledu a obsahu dokumentu
- Řízení prohlížeče
- Interakce s formuláři HTML
- Interakce s uživatelem – ovladač událostí
- Čtení a zápis klientského stavu pomoci cookie

- Spolupráce s Java Applety

JavaScript umožňuje práci s objektem dokumentu. Je možné tedy úplně změnit načtený dokument včetně jeho kaskádových stylů. Je možné taky řídit chod prohlížeče. Zobrazit nové okno a manipulovat s ním. Tato funkce je však často zneužívána reklamou na stránkách a proto bývá prohlížeči zakázána. Díky možností sledovat různé události na stránce je JavaScript často využíván například k validaci formulářů. Uveďme si také ale, na co je JavaScript krátký.

2.4.3 Omezení

- Z bezpečnostních důvodů JavaScript neumožňuje práci se soubory
- Neumožňuje práci se sítí

2.4.4 Vlastnosti jazyka

Dynamické přiřazení typů - Datové typy jsou asociovány s hodnotami namísto proměnných. Tzn. že proměnné může být přiřazena hodnota reálného čísla, která je později použita jako řetězec.

Objekty jako asociativní pole – Objekty jsou asociativními poli, rozšířenými o prototypy. Tedy k třídní proměnné „a“ se přistupuje jako k poli objekt[“a“], je ale také možné použít tečkové notace známé z jiných OO jazyků, tedy objekt.a.

Provádění příkazů, předávaných jako řetězce za běhu programu – JavaScript obsahuje metodu eval, která umožní např. spustit metodu, jejíž název není při psaní kódu ještě znám.

Funkce jako třída a konstruktor – Funkce jsou samy objektem. Můžou mít tedy vlastnosti a funkce, které jsou předávány dále. Parametry funkce jsou zároveň použity jako parametry konstruktoru. K vlastnostem objektu se přistupuje klíčovým slovem this.

Anonymní funkce – Velice často používaný prvek JavaScriptu. Funkce mohou být definovány uvnitř jiných funkcí a jsou vytvořeny při každém zavolání jejich nadřazené funkce. Mají přístup k proměnným funkce nadřazené a to i po ukončení volání nadřazené funkce.

Prototypy – Jak bylo řečeno, JavaScript má jen základní vlastností OOP. Nedisponuje konceptem třída-instance, jak ho známe z ostatních OOP jazyků. Existuje ale mechanismus prototypů, který rysy OOP částečně nahrazuje.

Výjimky – JavaScript umožňuje zachytávat chybové stavy mechanismem výjimek.

Nyní si ukážeme alespoň některé z vlastností na krátkém kódu.

```
function PrumerKruhu() { return this.r * 2; } // metoda třídy

function Kruh(x,y,r) // kontruktor
{
      this.x = x; // vlastnosti třídy
      this.y = y;
      this.r = r;

      this.prumer = PrumerKruhu; // metoda třídy
}

Kruh.prototype.PI = 3.14; // objekt prototypu, ze kterého třída dědí
Kruh.prototype.obvod = function() { return 2*this.PI*this.r };

var k = new Kruh(1.0,1.0,2.0); // vytvoření instance
var d = k.prumer; // metoda třídy
var o = k.obvod(); // metoda rodičovské třídy
```

2.4.5 Javascriptové frameworky

Samotný JavaScript nabízí poměrně zajímavé možnosti. Proč ho ale ještě více nezpestřit. A tak se v posledních letech začaly objevovat různé frameworky, které nabízí ještě pestřejší nabídku schopností. Proč tyto frameworky používat?

- Nabízejí grafické efekty a animace, které zpříjemní ovládání stránek
- Obsahují pokročilé uživatelské prvky jako našeptávače, dialogová okna apod.
- Registrace a obsluha událostí
- Hotová řešení ajaxu

Pokud vyžadujeme některé z těchto prvků, tak není nad čím váhat a framework použít. Pokud však zvažujeme jestli něco z výhod použít, či ne, je třeba se zamyslet i nad nevýhodami.

- Velikost frameworků – Frameworky jsou nutné načíst minimálně při prvním spouštění stránky. Jejich velikost je však nutné zvážit, i když s narůstají rychlostí připojení k internetu, se nejedná o tak velkou překážku.
- Rychlost – Použití frameworku je logicky pomalejší, než použití nativních volání
- Nutnost učit se nové rozhraní
- Ne vždy obsahuje jeden framework vše co potřebujeme a je nutné použít vícero druhů

Uveďme si alespoň několik nejznámějších frameworků současného trhu.

2.4.6 Prototype

Jeden z nejrozšířenějších a zároveň nejstarší z frameworků vůbec. První verze vyšla v roce 2005. Původ nachází v Ruby on Rails, kde sloužil hlavně pro ajaxovou komunikaci. Je postaven na rozšiřování JavaScriptu samotného. Má také vlastní vybavení pro podporu OOP v JavaScriptu.

2.4.7 YUI

YUI neboli Yahoo! UI Library je framework, který vyvinuli a používají vývojáři firmy Yahoo. Poprvé byl představen v roce 2006.

2.4.8 MooTools

Framework, který je založen na zmíněném frameworku prototype. Je využíván především díky svému množství vizuálních efektů, ale i dalších užitečných funkcí.

2.5 Jquery

Tento framework byl použit i v této práci, a proto si ho přiblížíme více. Byl uveden v roce 2006, ale od té doby si získal velký podíl na trhu JavaScriptových frameworků. Vývoj zajišťují jednak dobrovolníci, ale i placení programátoři. Vyjmenujme si největší výhody tohoto frameworku.

- Velice dobrá architektura a dokumentace
- Velké množství knihoven a pluginů
- Komunita
- Velikost knihovny
- Licence – přístupný pod licencemi MIT a GNU GPL

První, čeho bychom si všimli při pohledu na kód jQuery je odlišný zápis, než u samotného JavaScriptu. Pro pohyb v DOM jsou určeny tzv. selektory.

```
var listItems = $("ul > li");
```

Znak '$' je prototyp jQuery. Je možné tedy použít i volání `jQuery("ul > li");`. Samotné $() tedy vytváří objekt jQuery. V příkladu si, ale hlavně všimněme zápisu „ul > li”. Jedná se o první možnost adresace pomocí selektoru CSS. Druhou možností je použití XPath. Stejný výraz by pak vypadal takto:

```
var listItems = $("ul/li");
```

Třetí možností je použít metod jQuery k procházení DOM. Velice důležitou metodou je v jQuery metoda ready, která obsluhuje událost domready. Je tedy volána ve chvíli sestavení DOM, ale ještě před voláním metody load, která byla zpravidla používána v klasickém JavaScriptu. Ukážeme si ji na dalším příkladu, kde je také vidět snadná obsluha událostí a použití anonymní metody.

```
$(document).ready( function() {
      $(“p”).click(function() {
            $("#list").slideDown(140);
      }
});
```

2.6 JSON

V této práci je také využito AJAXU. Tedy techniky pro asynchronní zasílání dat mezi JavaScriptem a serverem. Jednou z možností je zasílaná data upravit do notace JSON. JSON tedy není programovacím jazykem, či frameworkem, ale souhrnem pravidel, jak formulovat data. Jedná se o textově orientovaná data, která jsou jednoduchá jak pro čtení a zápis člověkem, tak pro parsování a generování počítačem. Základem jsou dva typy struktur. První je kolekce dvojic klíč-hodnota, které jsou v programovacích jazycích reprezentovány datovým typem slovník, či hašovací tabulkou. Druhou možností je uspořádaný seznam hodnot, což se dá reprezentovat datovým typem pole, vektor, list apod..

Všechna pravidla bychom našli na stránkách json www.json.org. Uveďme si alespoň krátký příklad, jak taková struktura může vypadat, a jak jí pomocí knihovny jQuery-JSON odeslat.

```
var menu = {
            page : $("#menu_item").attr("tmp"),
            right : $("#menu_rights_list option:selected").text()
        }
$.getJSON("scripts/ajax/menu.php", menu, saved);
```

Ke zpracování json v php jsou implementovány funkce json_decode a k odeslání json_encode. Podobné funkce bychom našli, ale i v mnoha dalších jazycích, takže je velice užitečný.

2.7 Systém pro tvorbu vzhledu

Probrali jsme si teorii používanou v systému a nyní si popíšeme, jak je systém navržen a jak funguje. První ze všeho jsem hledal možnost implementovat již nějaký existující šablonovací systém. Žádný však neodpovídal mým představám. Pomocí systému má být totiž

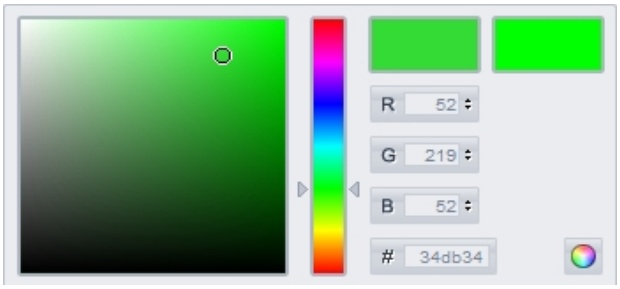
člověk bez znalosti programování schopen sestavit stránky. Takže by měly být všechny prvky ovládané skrz grafické rozhraní. Rozdělil jsem systém na několik částí:

- Správa barevných motivů
- Správa struktury stránky
- Správa struktury webu

Jako je v různých platformách základním kamenem třída Objekt, je v tomto systému základem jakýsi objekt Box. Úplně holý objekt není ničím jiným než elementem div. Je tedy vhodný jak pro strukturování stránky, tak pro grafickou úpravu pomocí kaskádových stylů. Jeho základní nastavení jsou

- Barevný motiv
- Tvar – Kulaté nebo hranaté rohy
- Možnost připojit stín
- Typ okraje – K dispozici jsou hned 4 možnosti
- Velikost
- Možnost nastavit ke každé instanci vlastní kaskádové styly
- Typ obsahu

Motiv obsahuje několik barev, které jsou nutné pro chod objektu Box. Pro editaci motivu je k dispozici aplikace využívající jQuery pluginu ColorPicker. Výběr barev tedy ve výsledku vypadá takto jednoduše:



Obrázek 2.1: ColorPicker, plugin jQuery

Aby bylo možné vytvořit další 4 body, byl vymyšlen mechanismus na generování obrázků okrajů a rohů. Navíc, jelikož existuje mnoho kombinací, jsou k nim vygenerovány i kaskádové styly. A pro další jejich použití, jsou veškeré informace ukládány do databáze.

Systém obsahuje sadu tříd, pomocí nichž lze vytvořit vlastní vzhledovou třídu. Jelikož se máme ale vyhnout jakýmkoli nárokům na programování uživatele, je několik takových tříd již připraveno.

Pro vytváření struktury stránky se tedy používají připravené vzhledové třídy, kterým uživatel nastavuje zmíněné vlastnosti. Aby bylo možné tyto třídy hierarchicky uspořádat a něčím je naplnit, je jim uživatelem zadán typ obsahu. Typů existují 4 druhy.

- Text – Obsah je statický text
- Aplikace – Obsah tvoří jedna z vybraných aplikací, které jsou v systému implementovány
- Layout – Logický kontejner, do kterého je možno zařadit další objekt boxu typu text, aplikace nebo layout
- Page – Význam je stejný jako u layout, ale takto definovaný objekt boxu je navíc možno zařadit do hiearchie stránek webu

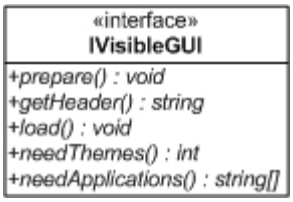
Po navržení struktury stránky editor vygeneruje potřebné obrázky, kaskádové styly a šablonu stránky. Vše je kvůli rychlé odezvě načtení uloženo na disk.

Teď už se dostáváme k poslednímu bodu vyřčenému na začátku kapitoly a tím je správa struktury webu. V této části uživatel vytváří strukturu webu pomocí stránek které si dříve vytvořil. Jednoduché ovládání, stejně jako u editoru stránky je implementováno pomocí knihovny interface frameworku jQuery. Informace o něm a hlavně jeho dokumentaci je možno nalézt na stránkách interface.eyecon.ro. Aby bylo možné některé stránky určitým uživatelům zneviditelnit, především jde o administrační nástroje, jsou stránkám nastaveny práva, které musí uživatel vlastnit. Výsledkem této správy je rovněž vygenerování struktury menu, která s tím souvisí. Přidáme si ještě malou ukázku, jak tento editor vypadá.

Obrázek 2.2: Ukázka aplikace pro strukturování webu

Na chodu menu bylo opět využito možností jQuery, konkrétně knihovny superfish. Tu je možné nalézt na adrese users.tpg.com.au/j_birch/plugins/superfish.

Téměř u každé z částí systému se bavíme o různých knihovnách a pluginech pro jQuery. Proto bylo taky třeba vyřešit jejich načítání až ve chvíli, kdy je to skutečně nutné. Proto aplikace, které systém obsahuje musí dědit rozhraní IvisibleGUI, které tedy nejen slouží pro načítání knihoven jQuery, ale také pro externí načtení kaskádových stylů a jakýchkoli dalších skriptů(getHeader()). Také je použito pro spuštění samotné aplikace, ve chvíli kdy je to třeba(load()). A taky obsahuje definice metod, které jsou nutné pro chod editoru stránky(needThemes(), needApplications()).



Ještě si uvedeme jakým způsobem jsou stránky zobrazovány. Řídící jednotkou je v tomto systému skript index.php. Podle parametru page, zjistí, zda daná stránka existuje, jaké je na ni nastaveno právo a podle toho buď pokračuje dál nebo se ukončí. Pokud je vše v pořádku, je nahrána šablona stránky. Podle šablony a pomocí zmiňovaného rozhraní se načtou potřebné aplikace a vzhled.

Tolik k systému tvorby vzhledu a přejděme k další části, kterou je fotogalerie.

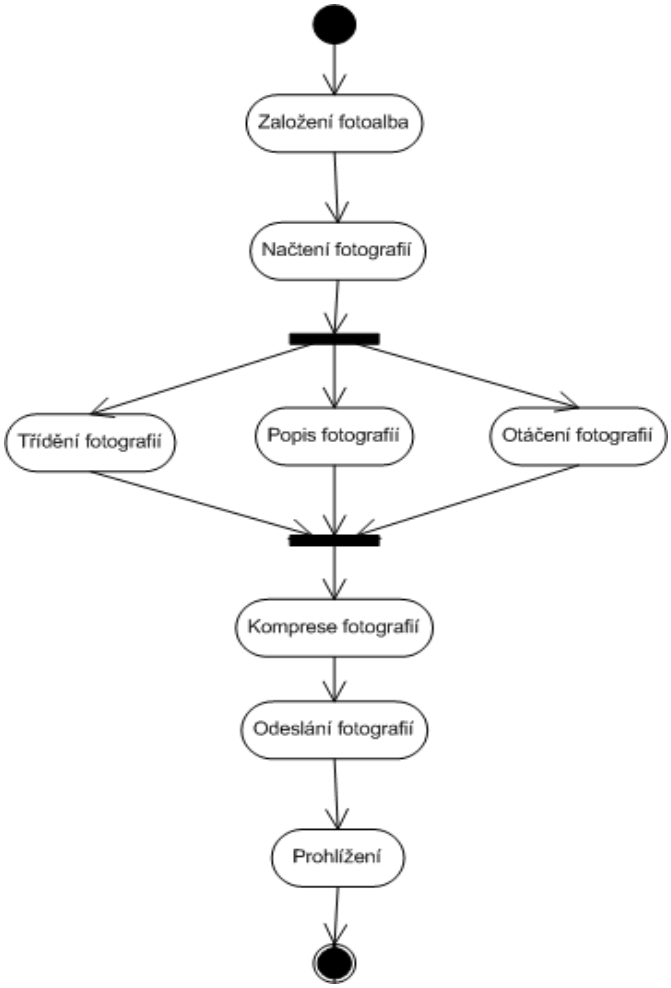
3 Fotogalerie

3.1 Požadavky

- Hromadné nahrávání fotografií
- Otáčení fotografií
- Komprese fotografií před odesláním na server
- Přívětivé a moderní uživatelské rozhraní pro prohlížení fotografií

3.2 Úvod

Nejprve si demonstrujme princip fotogalerie na jednoduchém UML diagramu aktivit.



Obrázek 3.1: Diagram aktivit fotogalerie

Při zpracování požadavků je nutné si uvědomit, kde se jednotlivé aktivity budou odehrávat. Zda na straně serveru, či na straně klienta. Jelikož je jedním z požadavků komprese fotografií před odesláním, budou se všechny aktivity od začátku až po kompresi odehrávat na straně klienta. Následné prohlížení fotografií se už bude odehrávat na straně serveru.

Nyní se zaměřme na výběr potřebných technologií. Prvním požadavkem je hromadné načtení fotografií. Nejjednodušším řešením by bylo využít JavaScriptu, ke kterému existují i volně dostupné knihovny, řešící toto téma. U aktivit otáčení a komprese fotografií však Javascript naráží na problém s nesnadnou práci s binárními daty.

Musíme tedy hledat jiné alternativy. Jednou z nabízejících se je vytvořit desktopovou aplikaci. Shrňme si její klady a zápory.

+ Lze vytvořit platformě nezávislou
+ Běží nezávisle na webovém prohlížeči
- Nutnost aplikaci instalovat
- Oddělenost od prohlížení fotografií

Další alternativou je vytvořit aplikaci, běžící přímo v prohlížeči. Uveďme si i zde klady a zápory.

+ Aplikace se nemusí instalovat
+ Běží v rámci celé aplikace v prohlížeči
+ Lze vytvořit platformě nezávislou
- Menší nabídka technologií
- Tyto aplikace mají jistá bezpečnostní omezení

Z uvažovaných možností byla zvolena aplikace bežící v rámci prohlížeče. Hlavním důvodem bylo kladení důrazu na neoddělování fotogalerie na více částí. Pro úplnost si uvedeme, že aplikací běžící v rámci jiné aplikace se říká applet. Ve stručnosti si projděme nejpoužívanější applety dnešní doby, které jsou schopny splnit požadavky pro nahrávání fotografií.

3.2.1 Adobe Flash

Asi vůbec nejrozšířenější multimediální platforma sloužící převážně pro tvorbu interaktivních animací, prezentací a her vkládaných do webových stránek. Byl vytvořen v roce 1996 firmou Macromedia. Dnes je však vyvíjen a distribuován firmou Adobe Systems.

Flash umí pracovat s vektorovou i rastrovou grafikou. Podporuje přenos zvuku a videa. Má vlastní objektově orientovaný programovací jazyk ActionScript. Aplikace jsou sestavovány do souborů swf. K jejich běhu je nutné mít však nainstalován Flash Player. Nicméně díky jeho velké rozšířenosti a integrace do webových prohlížečů je flash dnes velice využíván. Flash je

implementován jak pro počítačové operační systémy jako Windows, Linux, Solaris či Mac OS, tak pro mobilní jako Symbian či Windows Mobile.

Flash má však i své stinné stránky. Jeho proprietární charakter znepokojuje hlavně zastánce otevřených standardů a svobodného software. Také jeho bezpečnost donutila několik světových bezpečnostních expertů nedoporučit jeho používaní.

3.2.2 Microsoft Silverlight

Silverlight je další webovou platformou, která poskytuje podobnou funkcionalitu jako Flash. Integruje práci s grafikou, animacemi a interaktivitou do jednoho běhového prostředí. Oproti Flashi je však hodně mladý. Vůbec první verze byla vydána v roce 2006 a to za účelem streamování videa. Tato verze také používala pro aplikační logiku JavaScript.

Až pozdější verze přinesly větší interaktivnost. Dnes je navíc programovatelný ve všech jazycích platformy .NET. Silverlight poskytuje grafický systém podobný WPF. Uživatelské rozhraní je psáno v jazyce XAML.

Silverlight je na tom s podporou o něco hůře než Flash. Podporován je samozřejmě operační systém Windows. Dále pak Mac OS. Pro Linux vyvíjí firma Novell implementaci pod názvem Moonlight. Nicméně tato implementace bude pravděpodobně vždy o krok pozadu. Vydání implementace pro mobilní operační systémy Windows Mobile a Symbian je plánovaná v průběhu roku 2010.

3.3 Java Applet

Pro zpracování, komprimaci a odeslání fotografií byl nakonec použit Java Applet, a proto si ho popíšeme trochu hlouběji.

Nejedná se o nic jiného než speciální aplikaci nad platformou Java. Aplikace je zkompilovaná do Java bytecode a zabalená do archivu jar. Archivace však není nutnou podmínkou. Ke svému běhu potřebuje webový prohlížeč s Java Plug-in a JRE. Je možno ji také spustit pod programem AppletViewer. Ten je hlavně určen pro testování. Díky běhovému prostředí JRE je applet zcela nezávislý na operačním systému. Navíc tím, že je applet jen speciální Java aplikací, stačí jen drobné úpravy v kódu a z appletu se může stát desktopová aplikace. Pro představu si ukažme jak vypadá implementace úplně jednoduchého appletu, který vypíše na obrazovku text.

```
import javax.swing.JApplet;
import javax.swing.JLabel;
import javax.swing.SwingUtilities;

public class TestApplet extends JApplet
{
    @Override
    public void init()
    {
        try {
            SwingUtilities.invokeAndWait(new Runnable() {
                public void run()
                {
                    JLabel label = new JLabel("Jednoducha ukazka Java
Appletu");
                    add(label);
                }
            });
        } catch (Exception ex) {
            System.err.println("GUI se nepodařilo vytvořit.");
        }
    }
}
```

Jak bylo řečeno, jedná se jen o speciální Java aplikaci. Chceme-li vytvořit Java applet, musíme zdědit třídu Applet nebo JApplet. Pokud nepotřebujeme používat možnosti grafického rozhraní Swing, postačí zdědit třídu Applet. Jestli ale chceme plně využívat knihoven Swingu, je třeba zdědit třídu JApplet, čímž se však o nic neochudíme, jelikož tato třída stejně dědí ze třídy Applet.

Třída Applet poskytuje několik metod, které jsou volány v závislosti na nějaké události. Po zdědění této třídy tedy využijeme vlastností objektově orientovaného programování a metody můžeme přepsat. Jak vidíme na příkladu, přepsali jsme metodu init, která je volána ve chvíli, kdy je applet načten. Jelikož vytváříme uživatelské rozhraní, měli bychom to provádět skrz volání metody invokeAndWait třídy SwingUtilities. Více informací o této metodě je možno získat v dokumentaci Javy. V metodě init tedy vytvoříme popisek pomocí třídy JLabel, který metodou add přidáme na plochu appletu.

Uveďme si ještě další užitečné metody třídy JApplet. První užitečnou je metoda start. Ta najde využití, pokud je vyžadováno spuštění nějakého procesu po inicializaci. Další je metoda stop. Tu by měly používat applety, kde je používána metoda start. Měla by zastavit vykonávání appletu. Hodí se především k zastavení appletu, pokud není zrovna viditelný na stránce. Poslední je metoda destroy. Ta bývá mnohdy nahrazena metodou stop, která je volána před metodou destroy. Avšak tato metoda je především k dispozici, pokud je nutné uvolnit nahraná data.

Kód Java Appletu je většinou psán v programovacím jazyku Java. Je ale také možno použít dalších jazyků, které jdou zkompilovat do Java bytecode. Jejich počet je razantně menší než je tomu u platformy .NET, ale ta možnost existuje. V dnešní době jsou známy především Jython, Ruby či Eiffel.

Rychlost Java Appletu je srovnatelná, i když obecně pomalejší, s jiným kompilovanými jazyky jako např. C++. V porovnáním s JavaScriptem je jeho rychlost několikanásobně větší. Java Applet může využít i 3D akceleraci, která je dostupná na platformě Java. Díky tomu jsou Java Applety vhodné a používané i pro složitější výpočty a jejich vizualizaci.

V následujícím příkladu si ukážeme, jak jednoduše se applet spouští v prohlížeči.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>Ukázka spuštění Java Appletu</title>
    </head>
    <body>
        <applet
            code = "Program.Main"
            archive = "Applet.jar"
            width = 640
            height = 480
        >
        </applet>
    </body>
</html>
```

Do kódu html stačí jen přidat tag <applet>. Z jeho atributů si všimněme především prvních dvou. Atributem code určujeme název třídy, kterou je applet spouštěn. V tomto případě tedy třída Main v balíčku Program. Atribut archive označuje název archivu jar ve kterém je applet uložen. Zbylé dva atributy určují velikost okna, ve kterém je applet spuštěn.

Java Applet také umožňuje definovat vstupní parametry, tak jak je tomu u konzolových aplikací. Stačí přidat element <param> do tagu <applet>. V dalším příkladě si to ukážeme názorně. Přidáme appletu parametry přebírající řetězec znaků a celočíselnou hodnotu.

```
<applet
      code = "Program.Main"
      archive = "Applet.jar"
      width = 640
      height = 480
      >
      <param name="paramStr" value="Ukazkovy retezec"/>
      <param name="paramInt" value="100"/>
</applet>
```

K získání parametrů v kódu appletu je třeba jen zavolat metodu getParameter, které jako argument předáme název chtěného parametru.

Jednou z užitečných funkcí, kterou Java Applet disponuje je umění komunikovat s JavaScriptem v prohlížeči. Jelikož se v této práci zabýváme i JavaScriptem, ukážeme si jak se tato komunikace implementuje. Ze všeho nejdříve je třeba zkontrolovat, zda máme v classpath nastavenou cestu ke knihovně plugin.jar. Ta je součástí Java SDE a je možno ji nalézt v adresáři „<cesta k JDK>/jre/lib/". V této knihovně využijeme třídy netscape.javascript.JSObject. Nejprve si ukažme kód html stránky s JavaScriptem, který posléze vyuzžijeme.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
  <head>
    <title></title>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <script type="text/javascript">
        function getCount() {
            return 100;
        }

        function writeSomething(thing) {
            something = document.getElementById("something");
            something.innerHTML = thing;
        }
    </script>
  </head>
  <body>
      <applet
            code = "Program.Main"
            archive = "Applet.jar"
            width = 640
            height = 480
        >
        </applet>
      <p id="something" />
  </body>
</html>
```

V kódu si všimněme dvou funkcí JavaScriptu. Jedna jen vrátí nějaké číslo a druhá vloží text na stránku. To co nás v tuto chvíli zajímá, je ale především kód v appletu, který s tímto pracuje. Proto si ho ukažme.

```
import javax.swing.JApplet;
import netscape.javascript.JSObject;

public class TestApplet extends JApplet {
    public void start() {
        try {
            JSObject window = JSObject.getWindow(this);
            Number age = (Number) window.eval("getCount()");
            String something = "Muj vek je " + age;
            window.call("writeSomething", new Object[] {something});
        } catch (JSException jse) {
            jse.printStackTrace();
        }
    }
}
```

První ze všeho jsme získali metodou getWindow ukazatel na objekt JSObject. Zavolání funkce JavScriptu je pak jednoduché. Využijeme metody eval třídy JSObject, které jako parametr předáme název požadované funkce. V posledním kroku jsme potřebovali funkci předat argument. Pro tento účel existuje metoda call, které předáváme prvním parametrem název funkce a druhým pole typu Object, ve kterém jsou potřebné parametry.

Tato možnost volat funkce JavaScriptu je v některých situacích velice užitečná, ikdyž nesmíme opomenout, že Java Applet také dokáže komunikovat přímo s DOM stránky. Komunikace Javascriptu s Java Appletem je možná i opačně. Tedy JavaScript dokáže volat metody napsány v Javě. Může také ale přistupovat k třídním proměnným, pracovat s polem i vytvářet nové instance objektů.

3.3.1 Bezpečnost

Jelikož Java Applet může využívat veškerých možností platformy Java, mohlo by dojít po jeho spuštění k nežádoucím akcím, jako je přístup k datům na disku apod.. Proto je spuštěn v tzv. Sandboxu, což je bezpečnostní mechanismus, který odděluje běžící program. Aby ale mohly applety využít funkce, které Sandbox nedovolí, je možné applet podepsat certifikátem. Uveďme si jaké rozdíly jsou mezi podepsaným a nepodepsaným appletem.

Nepodepsané applety např. můžou:

- Uskutečnit síťové spojení se serverem, na kterém jsou spušteny
- Spustit veřejné metody ostatních appletů na stránce
- Číst bezpečnostní systémové proměnné (Secure System Properties)

Nepodepsané applety nemůžou:

- Přistupovat k datům klientského počítače

- Uskutečnit síťové spojení s jiným serverem, než ze kterého byly spuštěny
- Načítat nativní knihovny
- Číst některé systémové proměnné

Podepsané applety tyto omezení nemají a můžou zasahovat mimo Sandbox. U appletu pro práci s fotografiemi je nutné přistupovat na disk klientského počítače, abychom fotografie otevřeli. Proto je nutné applet podepsat a jak to udělat si ukážeme v další kapitole.

3.3.2 Podepsání Java Appletu

Abychom mohli applet podepsat, je nutné mít certifikát. K jeho vytvoření a správě slouží aplikace keytool, která je součástí platformy Java. Možnosti tohoto nástroje bychom našli v jeho dokumentaci. My si ale ukážeme jednoduchý příklad, jak vytvořit certifikát.

```
keytool -genkey -dname "cn=Michael Hrabalek, ou=UASFsoft, o=UASF, c=CZ"
      -keypass heslo -keystore C:\Klice
      -storepass klice -validity 180
```

Tímto příkazem se nejdříve vytvoří úložiště klíčů pojmenované Klice na disku C, které je zaheslováno heslem klice. Dále se vytvoří dvojice soukromého a veřejného klíče identifikovaných jménem Michael Hrabálek, organizační jednotkou UASFsoft, organizací UASF, označením země CZ, uložené pod heslem heslo. Nakonec je vytvořen certifikát, který obsahuje veřejný klíč a identifikační údaje. Poslední parametr -validity nastavuje platnost certifikátu. V tomto případě na 180 dní.

Když máme připraven certifikát, můžeme applet podepsat. K tomu slouží nástroj jarsigner, který je také součástí platformy Java. Jeho možnosti necháme opět na dokumentaci a ukážeme si jak jím lze applet jednoduše podepsat.

```
jarsigner -keystore C:\Klice -storepass klice
      -keypass heslo -signedjar SPhotoUploader.jar PhotoUploader.jar
```

V příkladu jsme nejprve určili, kde se nachází úložiště klíčů a hesla k přístupu do úložiště a certifikátu. Následně je definován název souboru, ve kterém bude podepsaný archiv, název souboru, který má být podepsán.

3.4 Applet pro zpracování fotografií

Nyní se zaměříme na vlastní applet, který byl v rámci práce vyvinut. Z úvodního diagramu aktivit je zhruba patrné, co applet obstarává, popíšeme si ale blíže jeho implementaci a problémy, které se při ní vyskytly.

Aby byly dodrženy pravidla dobrého programování, je aplikace rozdělena do dvou částí. Částí obstarávající vzhled a druhou vykonávající logiku programu. Vzhled je tvořen pomocí knihoven Swing. Bohužel při práci s obrázky tyto knihovny naráží na nepříliš rychlou odezvu, což je u aplikace pracující s fotografiemi zásadní problém.

Pro dobrou práci s fotografiemi jsou po jejich vybrání po celou dobu běhu aplikace k dispozici jejich náhledy. Je nemožné načítat celé fotografie, a proto jsou fotografie nejprve zmenšovány. K zobrazení náhledu je použita třída JToggleButton, která pro zobrazení využívá třídu ImageIcon. Jako první se tedy nabízelo využít metody getScaledInstance třídy Image. Při testování se však zmenšování fotografií tímto způsoben nezdálo příliš rychlé. Bylo tedy nutné hledat jiné východisko a provést testování rychlosti.

Po hledání v literatuře bylo navrženo nové řešení vytváření náhledů fotek a to pomocí tříd BufferedImage a Graphics2D.

```
BufferedImage thumbImage = new BufferedImage(width, height,
      BufferedImage.TYPE_INT_RGB);
Graphics2D graphics2D = thumbImage.createGraphics();
graphics2D.setRenderingHint(RenderingHints.KEY_RENDERING,
      RenderingHints.VALUE_RENDER_SPEED);
graphics2D.drawImage(image, 0, 0, width, height, null);
icon = new ImageIcon(thumbImage);
```

Uveďme si výsledky testování těchto dvou způsobů. Měření bylo prováděno s padesáti kusy fotografií, dohromady o velikosti 180Mb, na počítači s procesorem Core2Duo 1,6GHz a 2Gb paměti RAM. Také bylo vyzkoušeno vytváření náhledů současně na více vláknech. Pro metodu getScaledInstance třídy ImageIcon je použita zkratka GSI a pro druhé řešení zkratka G2D. Čas je uváděn v sekundách zaokrouhlen na desetiny.

	1 vlákno – GSI	**1 vlákno – G2D**	**2 vlákna – GSI**	**2 vlákna G2D**
1	72	40,9	46,2	27,2
2	72,2	40,9	46,3	27,2
3	72,1	41	46,4	27,2
4	72,3	40,9	46,2	27,2
5	72,1	40,9	46,3	27,3
Ar. průměr	**72,2**	**40,9**	**46,3**	**27,2**

Bylo také prováděno měření na více vláknech. To však bylo velice podobné číslům u vláken dvou. Jak je patrné z výsledků, oproti původní implementaci došlo k více než dvou a půl krát násobnému zrychlení.

Během načítání náhledů se vyskytl ještě jeden problém, který byl mnohem kritičtější, než rychlost načítání. Při otevření objemnější fotografie, zvláště pak při práci více vláken, tedy současné otevření více fotografií, JVM hlásil přetečení paměťového prostoru. JVM totiž vyhradí procesu appletu jen 128Mb paměti, což je pro načtení nekomprimované fotografie opravdu málo. K překvapivému zjištění však toto není uváděno v téměř žádné literatuře zabývající se výukou Java Appletů. Naštěstí je na tento problém vývojáři Javy pamatováno a velikost vyhrazené paměti se dá nastavit pomocí parametrů v kódu html pro spuštění appletu.

```
<applet
      code = "program.Main"
      archive = "PhotoUploader.jar"
      >
      <PARAM name="java_arguments" value="-Xmx256m">
</applet>
```

Takto bychom nastavili paměťový prostor na 256Mb.

Po načtení náhledů si uživatel může fotografie otáčet a třídit. Požadavkem na odeslání jsou fotografie postupně zmenšovány a komprimovány do dvou velikostí. Pro náhledy a do podoby vhodné k prohlížení. K otáčení a škálování je opět využito třídy BufferedImage. Komprimace využívá třídy ImageWriter. Ke komunikaci se serverem je pak použito požadavku POST a protokolu HTTP. Na serveru jsou fotografie zpracovány skriptem PHP, informace o nich jsou uloženy do databáze a samotné fotografie na disk.

3.4.1 Prezentace fotografií

Procházení fotogalerie je postaveno na komponentách vyvinutých při tvorbě systému pro tvorbu vzhledu. Jelikož je v této práci využito JavaScriptového frameworku jQuery a jelikož je v něm vyvinuto hned několik pěkných aplikací pro prezentaci fotografií, rozhodl jsem se použit jednu z nich. Jedná se o součást knihovny interface a nese název ImageBox.

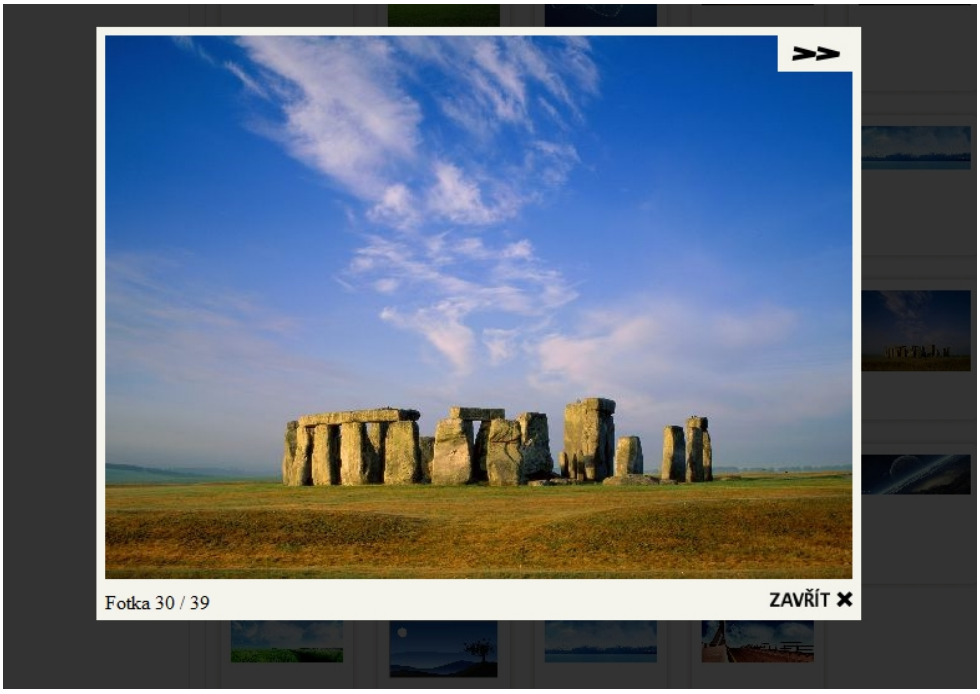
Její implementace k použití je jednoduchá. Potřebný objekt se obalí tagem <a>. Jako hodnota atributu href se nastaví cesta k souboru, do atributu title je možné vložit popis fotografie a všem fotografiím, které chceme zobrazit v jedné prezentaci nastavíme stejnou hodnotu atributu rel. Podmínkou je aby ale začínala slovem imagebox.

```
<a href="photos/a35/1.jpg" title="Loď" rel="imagebox-album">
<img id="a35/1" src="photos/a35/tmb/1.jpg" alt="photos/a35/tmb/1.jpg"
class="photka" /></a>
```

Dále je nutné si uložit a nastavit cestu k souboru kaskádových stylů nebo se můžeme pokusit napsat vlastní. Samotné spuštění aplikace v kódu javascriptu spočívá v zavolání metody

init třídy ImageBox. Na závěr si ještě ukažme jak ve výsledku vypadá Java Applet a prezentace fotografií.

Obrázek 3.2: Ukázka appletu pro nahráváni fotografií



Obrázek 3.3: Ukázka prezentace fotografií

4 Instalační aplikace

Jednou z variant provozu vyvíjeného systému je, že si ho uživatel bude moct stáhnout a spustit na svém hostingu. Je tedy nutné připravit jednoduchou aplikaci díky níž se podaří systém zprovoznit. Nutnou podmínkou pro instalaci je hosting s interpretem php a SŘBD MySQL. Kvůli SŘBD je tedy nutné aby při prvním spuštěny byly zadány do systému přihlašovací údaje k databázi, což je první aktivitou této aplikace. Další si uveďme na diagramu.

Po zadání údajů k databázi jsou do ní nahrány data potřebná pro další kroky. Díky vymyšlenému systému tvorby vzhledu jsou uživateli nabídnuta připravená rozvržení stránky a barevné motivy. Více už v instalační aplikaci není třeba, jelikož další potřebné funkce jako správa uživatelů, budou již nachystána ve vygenerovaném webu.

5 Uživatelská dokumentace

Uživatelská dokumentace je dle požadavků zpracována v samostatném dokumentu pdf, který je součástí aplikace a také jako příloha této práce.

6 Závěr

V této práci jsem se zabýval vývojem webové aplikace. Aplikace měla umožňovat také její instalaci u klienta. Jak vyplývá z názvu mělo se tedy jednat o systém, který je schopen především v každé jeho instanci jinak vypadat – být univerzální. Navíc měla být tato schopnost co nejvíce uživatelsky přívětivá, aby ji zvládl laik bez jakýchkoli znalosti IT, snad jen s pomocí stručné dokumentace.

Práce měla dvě hlavní části, čímž byl systém pro tvorbu vzhledu a fotogalerie. U obou částí jsou vysvětleny technologie, které jsou použity. V první části je dle zadání rozvinuto téma JavaScriptu a ve druhé části téma týkající se platformy Java, konkrétněji Java Appletu.

Během vývoje se vyskytlo mnoho problémů, které bylo nutné vyřešit. Některé z nich, zvláště ty, u kterých se řešení hledá hůře jsou v práci vysvětleny. Čtenář tak může získat cenné informace pro své vlastní potřeby.

V některých situacích bylo využito již existujících komponent. Jedná se především o knihovny jQuery. Jindy bylo nutné přijít s vlastním řešením, jako je tomu u systému tvorby vzhledu. U fotogalerie bylo zase nutné provést měření výkonu a navrhnout optimalizaci. Práce byla tedy poměrně komplexní, ale z hlediska vývoje zajímavá a poučná.

7 Literatura

Flanagan, David. Javascript Kompletní průvodce, 2. aktualizované vydání. Praha : Computer press, a.s., 2002. ISBN 80-7226-626-8

Javascript - Wikipedie, otevřená encyklopedie. Wikipedia. http://cs.wikipedia.org/wiki/JavaScript

Meyer, A. Eric. Eric Mayer o CSS – Kompletní průvodce. Brno : ZONER software, s.r.o., 2007. ISBN 978-80-86815-64-0

Herout, Pavel. Java – grafické uživatelské rozhraní a čeština. České Budějovice : Kopp, 2006. ISBN 80-7232-237-0

Darwin, F. Ian. Java Kuchařka programátora. Brno : Computer press, a.s., 2006. ISBN 80-251-0944-5

Teague, Jason Cradford. DHTML a CSS pro World Wide Web – praktická vizuální příručka. Praha : SoftPress, s.r.o., 2005. ISBN 80-86497-77-1

Resig, John. JavaScript a Ajax – Moderní programování webových aplikací. Brno : Computer press, a.s., 2007. ISBN 978-80-251-1824-5

McFarland, David Sawyer. CSS chybějící manuál. Praha : Grada Publishing, a.s., 2007. ISBN 979-80-247-2122-4

Schlossnagle, George. Pokročilé programování v PHP 5. Brno :ZONER software, s.r.o., 2004. ISBN 80-86815-14-5

Vondrák, Ivo. Úvod do softwarového inženýrství. Ostrava. 2002

Boumphrey, Frank. XHTML : Průvodce vývojáře. Praha : Mobil Media. 2002. ISBN 80-86593-14-2

Castro, Elizabeth. HTML 4 pro World Wide Web. Brno : SoftPress, s.r.o., 2001. ISBN 80-86497-08-9

Lennon, Joe. Compare JavaScript frameworks. http://www.ibm.com/developerworks/web/library/wa-jsframeworks/

Sun Developer Network (SDN), Oracle. Applet documentation from the Java Tutorial. http://java.sun.com/docs/books/tutorial/deployment/applet/index.html

Swedberg, Carl a Chaffer, Jonathan. jQuery 1.4 Reference Guide. Birmingham : Packt Publishing Ltd., 2010. ISBN 978-1-84951-004-2

8 Přílohy

8.1 Obsah přiloženého CD

Adresář	Obsah
/Text/	Text bakalářské práce
/Uasf/	Implementovaný systém
/Applet/	Podepsaný jar archív s appletem pro nahrávání fotografií
/Applet/PhotoUploader	NetBeans projekt s appletem pro nahrávání fotografií
/Instalace/	Archív se systémem připraveným k instalaci
/Uzivatel/	Uživatelská dokumentace